Network Service Dependencies in Commodity Internet-of-Things Devices

Poonam Yadav, Qi Li, Richard Mortier
University of Cambridge
Cambridge, UK
first.last@cl.cam.ac.uk

Anthony Brown
University of Nottingham
Nottingham, UK
first.last@nottingham.ac.uk

ABSTRACT

We continue to see increasingly widespread deployment of IoT devices, with apparent intent to embed them in our built environment likely to accelerate if smart city and related programmes succeed. In this paper we are concerned with the ways in which current generation IoT devices are being designed in terms of their ill-considered dependencies on network connectivity and services. Our hope is to provide evidence that such dependencies need to be better thought through in design, and better documented in implementation so that those responsible for deploying these devices can be properly informed as to the impact of device deployment (at scale) on infrastructure resilience. We believe this will be particularly relevant as we feel that commodity IoT devices are likely to be commonly used to retrofit "smart" capabilities to existing buildings, particularly domestic buildings.

To the existing body of work on network-level behaviour of IoT devices, we add (*i*) a protocol-level breakdown and analysis of periodicity, (*ii*) an exploration of the service and infrastructure dependencies that will implicitly be taken in "smart" environments when IoT devices are deployed, and (*iii*) examination of the robustness of device operation when connectivity is disrupted. We find that many devices make use of services distributed across the planet and thus appear dependent on the global network infrastructure even when carrying out purely local actions. Some devices cease to operate properly without network connectivity (even where their behaviour appears, on the face of it, to require only local information, e.g., the Hive thermostat). Further, they exhibit quite different network behaviours, typically involving significantly more traffic and possibly use of otherwise unobserved protocols, when connectivity is recovered after some disruption.

CCS CONCEPTS

• **Networks → Network measurement**;

KEYWORDS

IoT, Infrastructure, Measurement, Scalability, Resilience

ACM Reference Format:
Poonam Yadav, Qi Li, Richard Mortier and Anthony Brown. 2019. Network Service Dependencies in Commodity Internet-of-Things Devices . In *IoTDI '19: Internet of Things Design and Implementation, April 15–18, 2019, Montreal, QC, Canada.* ACM, New York, NY, USA, 11 pages. *https://doi.org/10.1145/3302505.3310082*

1 INTRODUCTION

It is widely believed that the number of Internet of Things (IoT) devices is growing rapidly and will exceed 20 billion by 2020 [16, 25, 33]. A large part of this future growth is expected to come from sensors, actuators and computation deployed in the built environment. Governments, commercial organisations, and private citizens are all experimenting with how IoT devices can make us, our cities, and our infrastructure more efficient. Standards such as Building Information Modelling (BIM) level 2 are increasingly widely used to model building design and construction, and future iterations (anticipated BIM levels 3 and 4) are expected to cover development of the built environment and associated infrastructure (e.g., transport, refuse, utilities, communications, health, education) more broadly [43]. For example, the UK Government has required "fully collaborative 3D BIM (with all project and asset information, documentation and data being electronic) as a minimum" since 2016.[1] Digitisation of our built infrastructure looks set to accelerate.

We focus here on domestic contexts: smart homes where typical IoT devices might include environmental sensors, security cameras, personal health and wearable devices, voice controlled assistants, and robots. Other contexts seem likely to contain distinct but overlapping devices. For example, smart hospitals might have rather more wearable health monitoring and medical devices (drug monitoring and delivery systems, pacemakers, etc.), and might perhaps integrate smart medical robots to provide efficient end-to-end workflow in the hospital [51]. In contrast, smart offices may share versions of a number of smart home devices for environmental sensing and control, while adding systems targeting the shared workspace to provide features such as online resource and space booking. Smart cities will include all of the above and will add various infrastructure sensing and control systems for lighting, parking, refuse collection, traffic control, and so on. All involve both local and centralised processing of information with complex data and information flow among heterogeneous components, implying dependencies on a wide range of network services and protocols [18].

The implications of this increased digitisation are not entirely clear however. For example, recent data breaches have continued

IoTDI '19, April 15–18, 2019, Montreal, QC, Canada

ACM ISBN 978-1-4503-6283-2/19/04.
https://doi.org/10.1145/3302505.3310082

[1] UK Government BIM level 2 mandate, *http://bim-level2.org/en/faqs/*

to increase sensitivity to the potential impact on security and privacy of widespread sensing, and a number of authors have examined the network bandwidth implications of widespread IoT deployment. In this paper we focus on a slightly different question: what are the implications on our built environment in terms of resilience and robustness if we come to rely on Internet-connected IoT devices? We begin to address this question by analysing data collected from lab-based measurement of the behaviour and service dependency of a range of domestic IoT devices covering different application domains, manufacturers, and popularity (§2). We analyse the collected data to understand traffic patterns, protocol usage, and service dependencies for the devices we monitor (§3). We then look specifically at the robustness of these devices under network disruption, examining how they respond to different types of interruption to their connectivity (§4). Finally we put our study in context (§5) and conclude (§6).

2 METHODOLOGY

To begin to understand the data types, rates, and traffic patterns caused by different IoT devices, we deployed a set of commodity off-the-self IoT devices in a small test area in an office in our lab, and captured the Wi-Fi traffic generated by these devices. Other occupants of the office were notified that the devices were present, and we carefully did not analyse the data captured from the devices for anything other than its gross network characteristics. The data captured thus represents a "minimum" level of traffic, and so service dependency, as the devices were not interacted with as they might be in a more realistic deployment. Our aim in this work is to uncover baseline data about network and service dependency of a selection of devices rather than to study device behaviour "in the wild". We certainly hope to explore device behaviour of more devices in more realistic deployment scenarios in the future but that is not the subject of this paper.

Table 1 describes the commodity IoT devices we deployed. All were connected to a local Netgear N600 Wireless Dual Band Router WNDR3700v2 running Linux OpenWrt version 2.6.39.4 [36] either wirelessly over standard 802.11b Wi-Fi or via an Ethernet cable, allowing us to capture all traffic to and from each device. Figure 1 depicts the experimental setup. We also made a very simple measurement of their energy consumption by connecting each device to a TP-Link Smart Plug for a fixed interval and recording the mean power consumption.

We categorise each device in one of two categories: (*i*) **Hub** refers to IoT devices which discover and control other IoT devices; (*ii*) **Sensor** refers to IoT devices which connect to the router and then communicate directly with various cloud services without using any Hub.

To collect data from all IoT devices in our network, we ran *monitor* and *collect* scripts on the router. We first get a list of MAC addresses of the devices. On the router, we run the *monitor* script on the interface that provides Wi-Fi access to the devices, filtering the traffic based on the MAC addresses of interest using *tcpdump* [44]. On the other side, we schedule a cron job to periodically upload collected data to a local development machine for offline processing. This was necessary both because processing on the router would have been too slow given its limited processing capacity, and because the router has limited persistent storage, less than 50 MB. Finally, we analyse all the *pcap* files, focusing on packet headers and control protocols such as DNS. For detailed packet analysis, we used Wireshark [48], GraphViz [20] and a set of custom Python scripts.

The result was a dataset spanning 8th March, 2018 to 11th April, 2018 although, due to an unobserved device failure, the D-Link Motion Sensor ceased interacting over HTTPS after just one day, and ceased network activity altogether after just less than one week; hence we only have data for that device from 4th April, 2018 to 10th April, 2018.

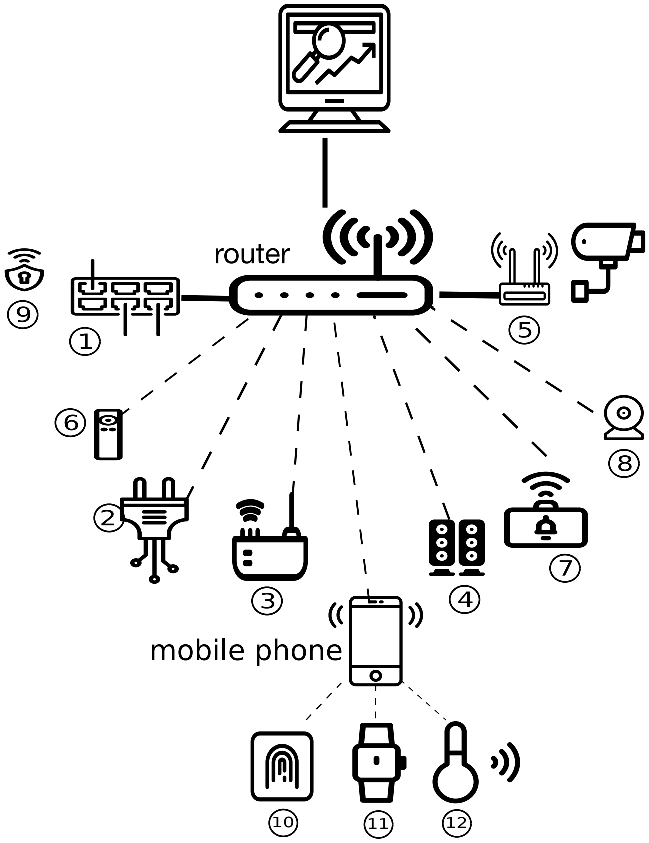


Figure 1: The Hive Hub (1) and Arlo Security Camera Hub (5) connect via wired Ethernet; the Hive Motion Sensor (9) communicates with the Hive Hub using Zigbee, and the Security Camera connects over Wi-Fi to the Arlo Security Camera Hub. Devices 10, 11, 12 connect to the smart phone over Bluetooth, and the rest (2, 3, 4, 6, 7, 8 and the controlling smart phone) connect over Wi-Fi. The router reaches the Internet via a wired Ethernet connection to the University's network. Detailed device descriptions are given in Table 1.

	Device	**H**ub/	Link Type	Protocols	**S**ecure/	Energy	Avg. Bandwidth
		Sensor			**I**nsecure	(W)	(B/s)
1	Hive Starter Kit Hub [23]	H	Ethernet	TCP, IGMP, ICMP	S	1.8	120
2	TP-link Smart Plug [46]	H	Wi-Fi	UDP, TCP	S,I	2.05	100
3	Google Home Mini [19]	H	Wi-Fi	UDP, TCP, IGMP, ICMP	S,I	1.4	125
4	Amazon Echo Dot [5]	H	Wi-Fi	UDP, TCP, ICMP	S,I	1.95	125
5	Arlo Security Camera Base Station [10]	H	Ethernet, Wi-Fi	UDP, TCP	S,I	4.6	70
6	Foobot Air Quality Monitor [15]	S	Wi-Fi	TCP	S	1.79	18
7	Nest Smoke Alarm [35]	S	Wi-Fi	UDP, TCP	S,I	NA	0.02
8	D-Link Motion Sensor [12]	S	Wi-Fi	UDP, TCP, IGMP	S,I	1.4	NA
9	Hive Motion Sensor [24]	S	Zigbee	HA 1.2	S	Battery	NA
10	ParrotPot Smart Flower Pot [38]	S	Bluetooth	V4.0 BLE	S	Battery	NA
11	MiBand Smart Bracelet [49]	S	Bluetooth	V4.0	S	Battery	NA
12	Smart Bluetooth Tracker [45]	S	Bluetooth	V4.0	S	Battery	NA

Table 1: IoT devices and their traffic behaviour summary.

3 ANALYSIS

There are many different ways to understand the network communications behaviour of devices. In this section, we present analyses of the traffic we collected from the setup shown in Figure 1, as that traffic was transmitted and received by the devices listed in Table 1. Our purpose is not to make generalised statements about all IoT devices but to illustrate some of the ways commodity devices behave and to consider the implications of those behaviours. We particularly look at the implications for service dependency and device robustness in subsequent sections.

3.1 Protocol Breakdown

Figure 2 presents a breakdown of the entire month's dataset by application protocol (Figure 2a), and per device by network and application protocol (Figures 2b and 2c). It is surprising to observe how much NTP, DNS and mDNS is in use by two devices in particular, the Smart Plug and D-Link Motion Sensor. It is also interesting to observe that only one device makes significant use of a classical IoT protocol (MQTT, used by the Foobot), though the Nest device also uses an IoT specific protocol (Weave) that was proprietary until released into Nest's developer platform in 2015. The rest use standard web protocols such as HTTP and HTTPS.

Local Network. For pairing and device discovery, many IoT hubs use low power and short range communication protocols to connect to devices (sensors). These protocols include Zigbee (IEEE 802.15.4) [3], Lora [41], Zwave [50], Lightwave [31], Bluetooth [21], RFID communication (LF (125–134 kHz), HF (13.56 MHz), UHF (433, and 860–960 MHz)) [40]. In our setup we have a few devices directly connecting to Hub using Zigbee, Bluetooth and Wi-Fi, e.g., the Hive motion sensor connects to the Hive Hub using the Zigbee protocol. We have three sensor devices which communicate with smartphone apps using Bluetooth and then those apps communicate with cloud services over the smartphone's Wi-Fi connection via the router.

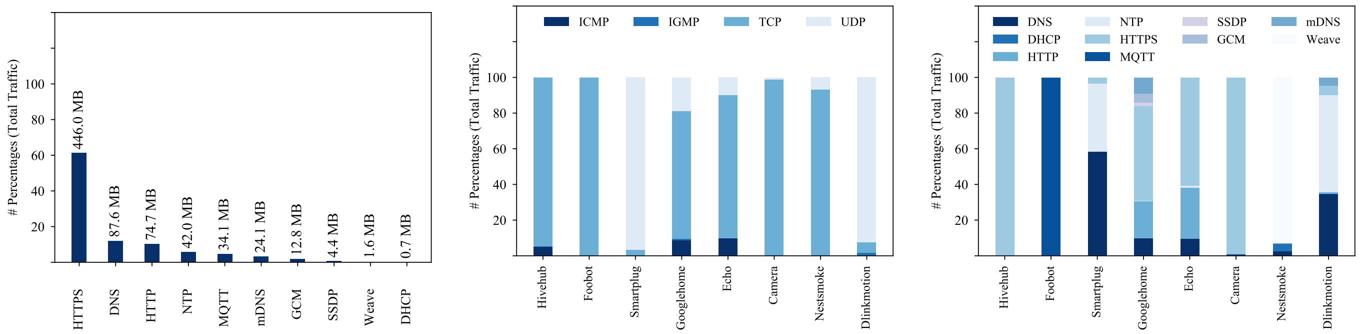
Encrypted Traffic. One straightforward observation we can make from the collected traffic traces is to observe the use of secure communication protocols between IoT devices and the outside world [17]. Figure 2c categorises traffic generated from each device based on application layer protocols. All IoT devices send at least part of their traffic using HTTPS, with **Hub** devices sending more (>50%) compared to **Sensor** devices. However we have not yet investigated how secure is the use of HTTPS by different devices in terms of selection of appropriate encryption suites and TLS configurations.

3.2 Traffic Characterisation

Figure 3 shows bandwidth consumption by considering aggregate bytes transmitted and received in 15 minute windows over one week.

The upward spikes shown for the Hive Hub (Figure 3a), Smart Plug (Figure 3c) and Security Camera (Figure 3f) are caused by devices receiving software updates, while the downward spikes are due to network congestion at the local router due to a Google Home software update early in the morning of 6th April (Figure 3d).

Amazon Echo (Figure 3e), Nest Smoke Alarm (Figure 3g) and D-Link Motion sensor (Figure 3h) showed some periodic upward spikes associated with frequent software updates. A high spike in the Security Camera Hub (Figure 3f) is due to the camera image-capture triggered by its motion sensor. In summary, we found > 99% of the Hive Hub's (Figure 3a) and Security Camera Hub's (Figure 3f) total traffic is composed of HTTPS packets and the remaining traffic comprises a few periodic DHCP, NTP and DNS interactions.



(a) Total traffic captured, by application. **(b) Total traffic per device, by network protocol.** **(c) Total traffic per device, by application.**

Figure 2: Traffic breakdown by protocol and device.

The majority of Foobot's (Figure 3b) traffic consists of MQTT running over TCP. Some of the devices, e.g., Smart Plug (Figure 3c), Amazon Echo (Figure 3e) and D-Link Motion sensor (Figure 3h), send frequent NTP traffic, and we see it forms a significant percentage of total traffic sent by these devices. As compared to all other devices, the traffic rate generated by Nest Smoke Alarm is small when it is in its ideal listening mode; it sends just 6 packets per day (totalling around 180 bytes per day). Nest Smoke Alarm uses the *Weave* protocol over TCP to communicate twice a day to the Nest Cloud Service.

We then analysed traffic from each device for periodicity across all active protocols during one week, 3rd April 2018 to 10th April 2018. We applied Discrete Frequency Fourier Transform (FFT) to detect periodicity in network duty cycle, and we present normalised the FFT magnitude vs frequency (cycles/week) in Figure 4. Our devices appear to all use DNS and DHCP periodically with other protocols typically used more intermittently.

We see that the Hive Hub uses only three application protocols (Figure 4a) with periodicity of 28 cycles/week for DHCP traffic, an average of 4 DHCP requests every day. DNS traffic shows different periodic cycles 7, 14, 21, 28, 35, 42, 49 cycles/week. It means some DNS requests frequency were in the range of 1 request per day to 7 requests per day. The Foobot network usage is aperiodic for three protocols with only DHCP showing periodicity at 32 cycles/week, i.e., 4–5 DHCP requests/day (Figure 4b). Out of three protocols used by Smart Plug, we see periodic behaviour with DHCP with two clear peaks, 24 and 28 cycles/week. However, there is some cyclic behaviour in DNS which means the device is making many DNS requests with periodicity 7, 14, 21, 35, 42 cycles/week as well (Figure 4c). Google Home uses seven application layer protocols and found four of them show some periodic behaviour. There are continuous NTP requests with a periodicity of 7, 14, 21, 35, 42, 49 cycles/week. DHCP period is strongest at 28 cycles/week, but shows smaller peaks for 7, 14, 21, 35, 42 (Figure 4d).

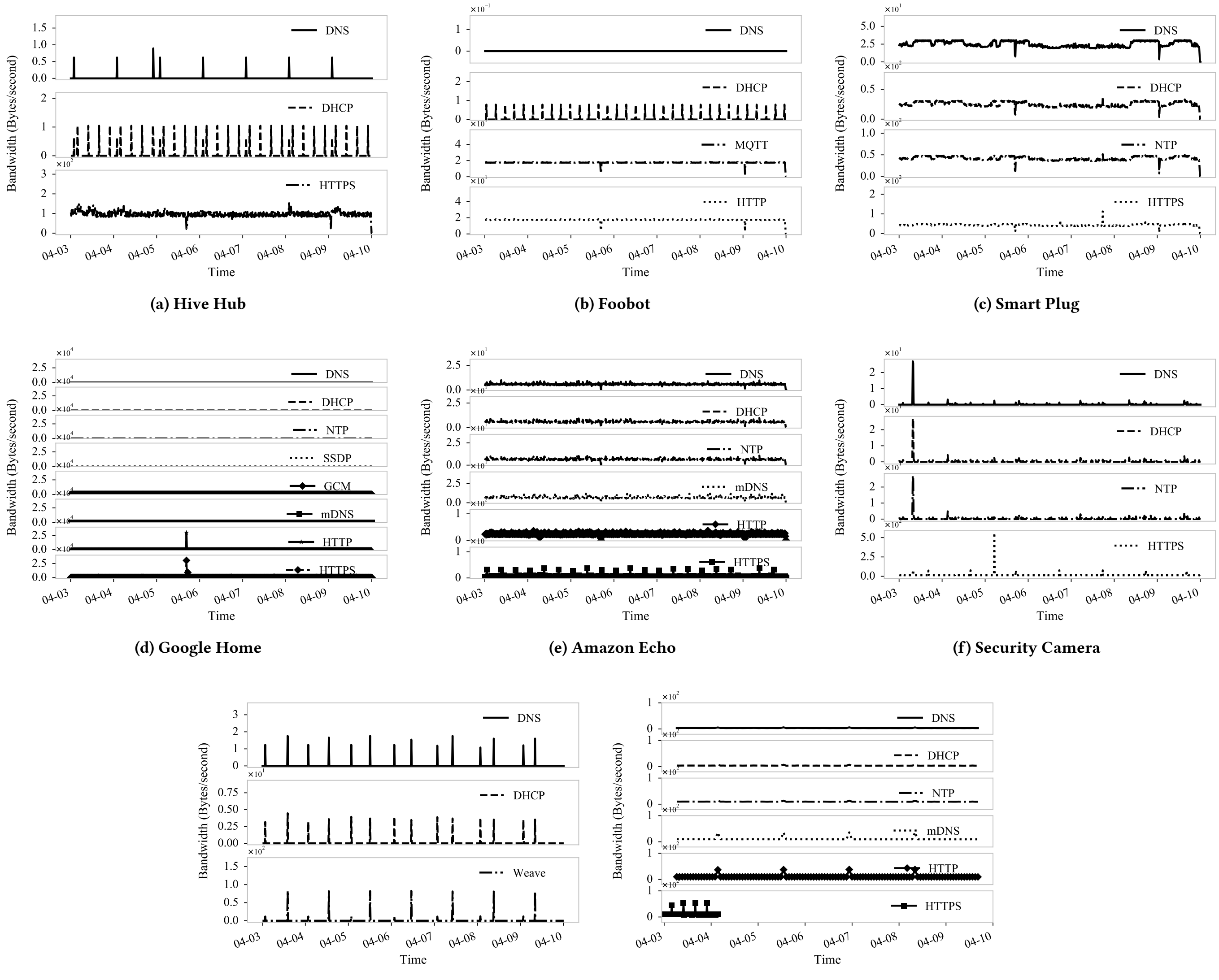
We also observed activity for GCM (Google Cloud Messaging) and HTTPS (analysed in the next section) at 1–2 cycles/week. Amazon Echo traffic composed of six protocols and DHCP and mDNS showed clear periodicity pattern at 28 cycles/week whereas HTTPS traffic showed periodicity 2–4 and 28 cycles/week in Figure 4e. Security Camera Hub showed clear periodicity for DHCP 18 cycles /week and various frequency peaks in DNS and NTP traffic as shown in Figure 4f. Nest Smoke Alarm showed periodic behaviour with all three protocols exhibiting frequencies 7, 14, 21, 28, 35, 42, 49 cycles/week (Figure 4g). D-Link Motion sensor showed periodic behaviour at 1 and 28 cycles/week, (Figure 4h). However, as noted above, this device suffered some unnoticed failure after just one day resulting in HTTPS traffic ceasing to be observed.

3.3 Protocol & Service Dependency

It is inevitable that IoT devices in deployment will depend on connectivity via a range of Internet protocols, both locally and to potentially many cloud-hosted services. We now examine some of these protocol and service dependencies for the devices we measured to illustrate some of the complex dependencies our infrastructure will take on if we increasingly deploy commodity IoT devices.

NTP usage in particular varied considerably between different devices, in terms of both the servers accessed and frequency. Some devices, e.g., the Hive Hub, used NTP during the setup process only. Some, e.g., the Foobot air quality monitor and Nest Smoke Alarm, use embedded timing protocols via MQTT and Weave rather than NTP. All those using NTP communicated with NTP servers run by the manufacturer except for the TP-Link SmartPlug which made extensive use of the global NTP Pool project servers [37].

Finally, we performed IP geolocation (using IP address allocation and routing data to infer the geographical location of a host with a particular IP address) to estimate the different countries hosting the services used by our devices. We used a command line utility on Linux platform, *geoiplookup*, to get the geolocation of a given IP address. For each combination of IoT device and application layer service, we aggregate the country names (also state names when the country is the USA) based on the ensemble of the IP addresses that the device communicates with. We filter out the NTP dependencies for Amazon Echo and Smart Plug as they each accessed tens of different locations due to their unusual usage pattern for NTP and DNS. We also extract some preliminary temporal information to describe the relative activeness of the communication: we divide the trace for each device into 15-minute windows



(a) Hive Hub

(b) Foobot

(c) Smart Plug

(d) Google Home

(e) Amazon Echo

(f) Security Camera

(g) Nest Smoke Alarm

(h) D-Link Motion

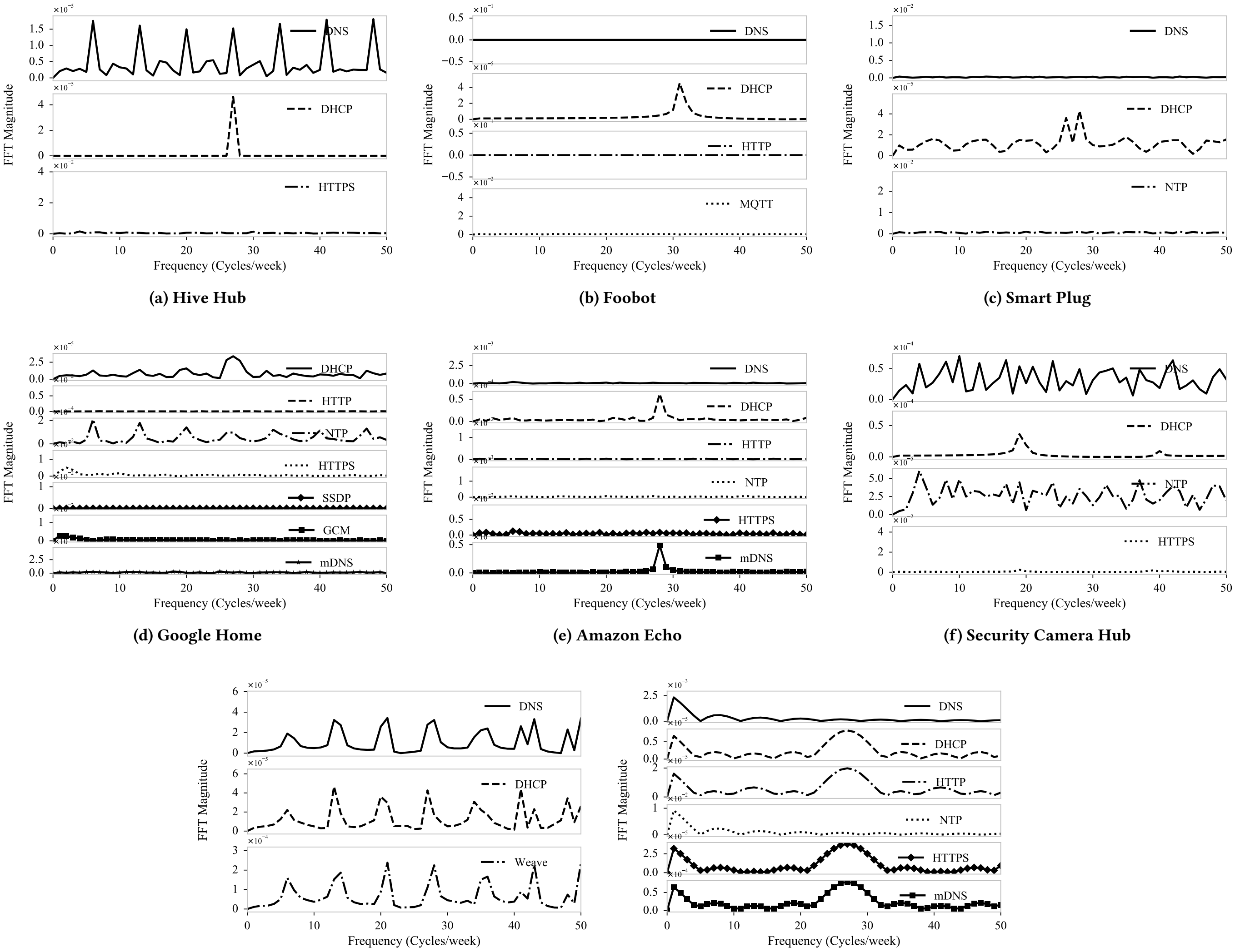
Figure 3: Per-device bandwidth used over one week, in 15 minute buckets. Note variation in y-axis scales. As noted above, an unobserved device failure truncated the D-Link Motion trace after one day for HTTPS and completely after one week.

and calculate the percentage of *active windows*, those where communication to/from a specific location actually happens. The results are shown in Figure 5.

Both the TP-Link Smart Plug and D-Link Motion Sensor make a large number of DNS queries to global NTP servers (Figure 6). The total DNS traffic generated by just 4 devices makes nearly 12% of *total* traffic generated from our setup. On the other hand, Hive Hub and Foobot air quality monitor make very few DNS queries to their servers. As we can see, most locations are quite inactive, with fewer than 10 locations where >50% windows are active.

	State	Internet	Local router	Devices
#1	Steady state	On	On	On
#2	Internet disconnected	Off	On	On
#3	Internet resumed	On	On	On
#4	Router power-off	Off	Off	On
#5	Router power-on	Off	On	On
#6	Internet resumed	On	On	On
#7	Device restarted	On	On	Off→On

Table 2: Experimental states use to examine effects of network disruption on devices.



(a) Hive Hub

(b) Foobot

(c) Smart Plug

(d) Google Home

(e) Amazon Echo

(f) Security Camera Hub

(g) Nest Smoke Alarm

(h) D-Link Motion

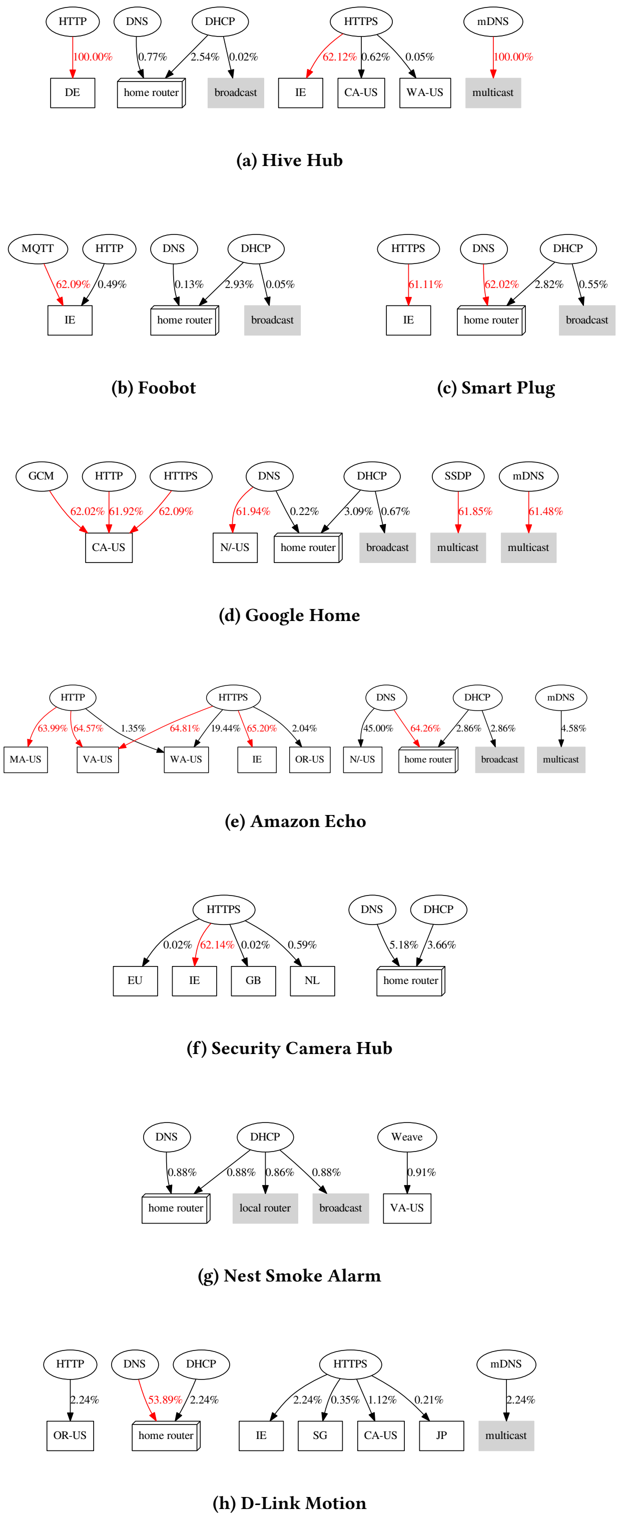
Figure 4: The periodicity of activity visible on the traffic generated for each device over one week by applying an FFT to the time-series data.

4 EFFECTS OF NETWORK DISRUPTION

Table 2 identifies seven different configurations for our experimental setup, each with different disruptions to device connectivity. We start with network and devices in steady state (#1), with all devices powered on, connected to the local router and thus to the Internet.

In our first experiment, we examine transitions from (#1) → (#2) → (#3), simulating Internet service interruption and recovery. We first cut off the Internet connection at the router by unplugging it from the University's network, and leave it unplugged for an hour to stabilise. In the meantime, we run scripts on the router to capture traces from the local network. We then reconnect the Internet connection to the router, and collect packet traces for a further hour.

Figure 2b shows the transport-layer traffic generated in steady state (#1), while Figures 7a show the equivalent after disconnecting the Internet (#2). After disconnection there is a significant increase in UDP traffic for all devices, while TCP traffic is reduced to zero for all except the Foobot. The UDP traffic is mostly composed of DNS queries. The Security Camera Hub significantly increased the amount of DNS (~30 kB) and DHCP (~10 kB) traffic in the following hour. All hub devices (Hive Hub, Google Home, Echo, and Camera Hub) also significantly increased their ICMP traffic, presumably attempting to diagnose the interruption and perhaps recover quickly when service is resumed; sensor devices continue to emit negligible ICMP traffic. Foobot and Echo are the only devices which transmit TCP traffic to the local router, Foobot's composed of MQTT and Echo's of HTTP. The Nest Smoke Alarm neither sent



(a) Hive Hub

(b) Foobot

(c) Smart Plug

(d) Google Home

(e) Amazon Echo

(f) Security Camera Hub

(g) Nest Smoke Alarm

(h) D-Link Motion

Figure 5: Geolocated service accesses by device.

nor received packets in this interval and so failed to detect lack of Internet connectivity.

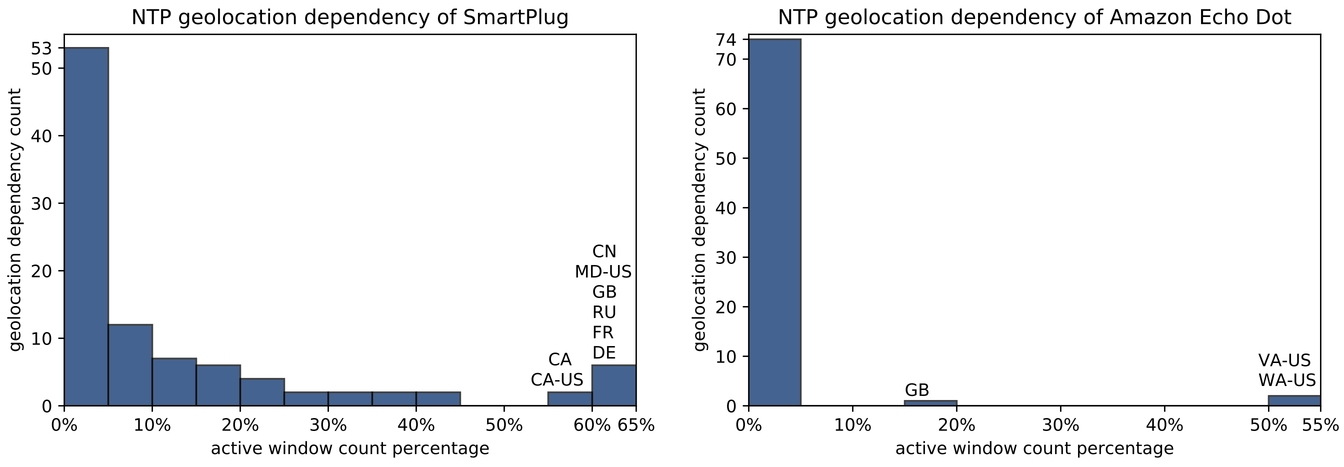
Figures 7c shows that only four application-layer protocols of the ten active in steady state were operational while there was no Internet connectivity (#2). During this period, only the TP-Link Smart Plug and Nest Smoke Alarm functioned as usual. Amazon Echo Dot and Google Home Mini responded to their wake words, but could only direct the user to check Internet connectivities. For all other devices, their respective apps showed them to be off-line, requiring user intervention. No devices seemed able to send HTTPS or HTTP traffic, severely limiting their functionality.

We then observe what happens when Internet connectivity resumes and we transition from (#2) to (#3). Figures 7b and 7d show the traffic generated when resuming. We see that only two of the four hubs (Hive Hub and Google Home) continue sending ICMP traffic, and only two devices (Hive Hub and D-Link Motion sensor) keep issuing IGMP requests. Five of the eight devices sent some TCP traffic in this state, suggesting that they detected connectivity had recovered. The Smart Plug and D-Link Motion sensors did not start sending any TCP but sent only DNS and NTP traffic, perhaps indicating both devices need to update global time before any TCP connection could be made. All devices continued to send UDP traffic, primarily DNS queries but the frequency reduced by $> 50\%$ compared to (#2).

Google Home only sent traffic composed of three Application layer protocols (MQTT, HTTP and DNS), showing that Google Home remains partially functional in (#3) for at least an hour. Echo and Camera Hub showed similar behaviour, both devices stopped ICMP traffic and sent DNS and HTTPS traffic. The DNS traffic of Camera Hub is reduced to nearly ~1 kB from 35 kB per hour.

Figure 9 shows the results of our second experiment where we examine network transition from states (#3) ⟶ (#4) ⟶ (#5) ⟶ (#6) per Table 2: having removed Internet connectivity, we then power off the router (#4) and test functionality of the two surviving devices from the first experiment. As expected, the Smart Plug's app works whether or not Internet connectivity is available as it only relies on local connectivity, but Nest Smoke Alarm's mobile app shows nothing about this disrupted connectivity as the user can still trigger an alarm check on the device through its Bluetooth connection. After the check, we turn the router's power back on, (#4) ⟶ (#5). We schedule a script to start collecting network traces right after the system finishes booting. We again remain in that state for an hour before turning the Internet back on at the router, (#5) ⟶ (#6). We summarise Internet dependency of the devices in Table 3.

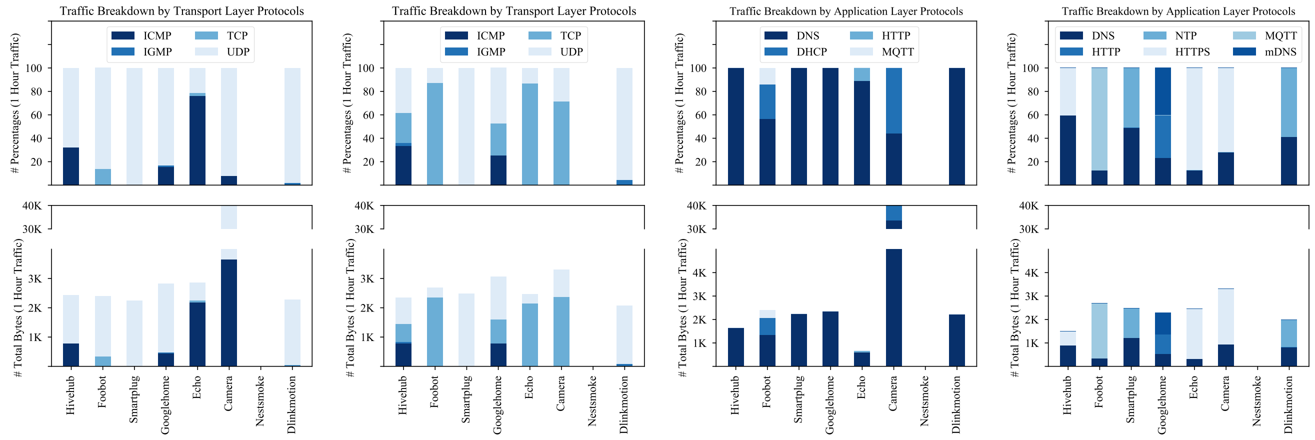
Figures 8a and 8b show the effect when the router is powered on but Internet is still disconnected, i.e., (#5). We see that only three transport-layer protocols (ICMP, IGMP and UDP) generate significant traffic in that hour, and there was no TCP traffic. All devices generated significant UDP traffic ($>$ 20 kB in first hour) as compared to (#2) which suggests current state behaviour depends on previous state. Google Home generated >250 kB of ICMP and UDP traffic with UDP traffic composed of DNS and SSDP. We found Echo behaved strangely after the router restart as it didn't detect the router automatically, instead connecting to our institution's open Wi-Fi network, resulting in missed traces from Echo.



(a) Smart Plug

(b) Amazon Echo Dot

Figure 6: Summary of locations accessed by the NTP services.



(a) (#1) → (#2)

(b) (#2) → (#3)

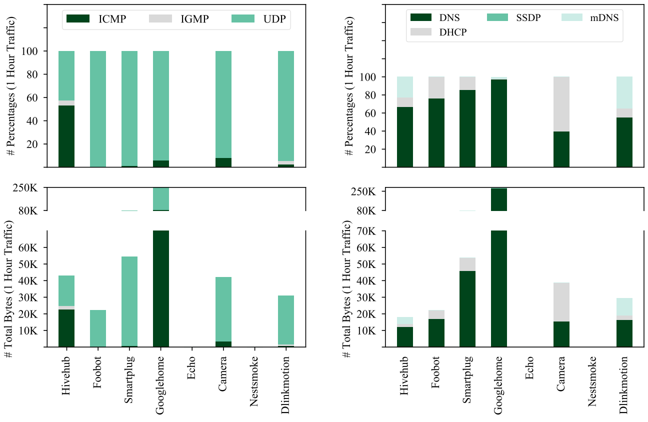
(c) (#1) → (#2)

(d) (#2) → (#3)

Figure 7: Traffic breakdown by transport- and application-layer Protocols for each device when network settings shown in Table 2 are changed from the state (#1) → (#2), and (#2) → (#3).

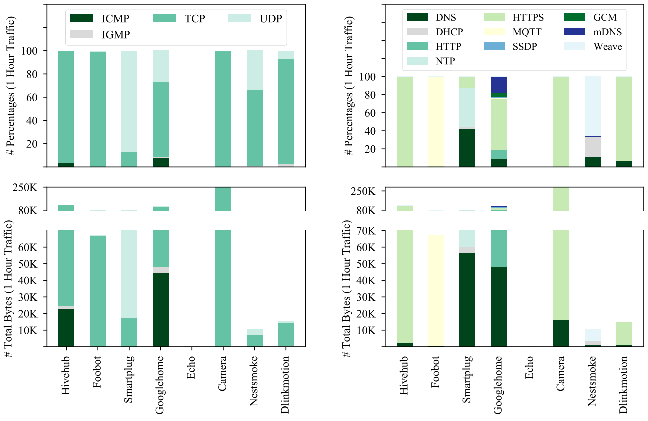
	Device	Functionality	Services Continued	Services Disrupted
1	Hive Starter Kit Hub [23]	Partial	DNS	DHCP, HTTPS
2	Foobot Air Quality Monitor [15]	Partial	DNS, MQTT, DHCP	HTTP
3	TP-Link Smart Plug [46]	Partial	DNS	NTP, HTTPS, DHCP
4	Google Home Mini [19]	Partial	DNS	HTTP, NTP, HTTPS, SSDP, GCM, mDNS
5	Amazon Echo Dot [5]	Partial	DNS, HTTP	NTP, HTTPS, DHCP, mDNS
6	Arlo Security Camera Hub [10]	Partial	DNS, DHCP	NTP, HTTPS
7	Nest Smoke Alarm [35]	Partial	DNS	Weave, DHCP
8	D-Link Motion Sensor [12]	Partial	DNS	NTP, DHCP, HTTP, HTTPS, mDNS

Table 3: Observed dependencies.



(a) Transport-layer traffic (b) Application-layer traffic

Figure 8: Traffic breakdown by transport- and application-layer protocols for each device when network settings shown in Table 2 are transition states, (#4) → (#5).

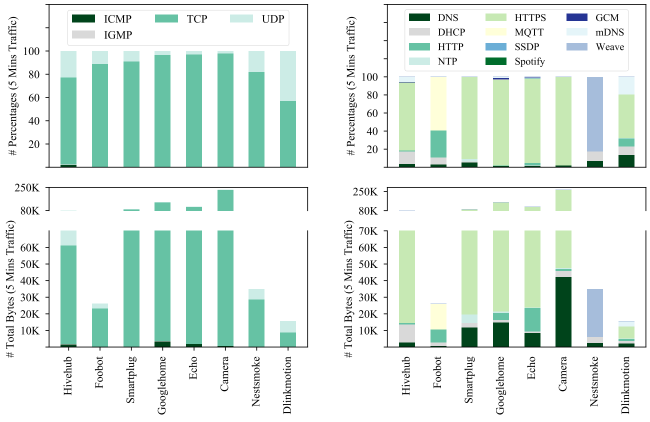


(a) Transport-layer traffic (b) Application-layer traffic

Figure 9: Traffic breakdown by transport- and application-layer protocols for each device when network settings shown in Table 2 are transition states, (#5) → (#6).

The effect of the transition from (#5) → (#6) is shown in Figures 9a and 9b where Internet connectivity is restored at the local router. We find that all ten application-layer protocols exhibit traffic, and all devices resume their normal functionality. Smart Plug (~55 kB), Google Home (~45 kB) and Camera Hub (~15 kB) sent significant DNS traffic in the first hour. The magnitude of total traffic generated by devices is nearly 10 times more than in state (#3), even though both states are superficially similar. This suggests that local router restart does briefly create a measurable increase in IoT traffic.

Figure 10 summarises traffic in the first 5 minutes after transitioning (#6) → (#7). In this state, devices are restarted one by one while the router remains powered on and Internet connected. After each device is restarted, we carried out one activity with it to ensure it was functioning normally. Each device was then kept idle for the rest of the measurement time. We analysed traffic from the 5 mins immediately after the device was restarted, and summarise this in Figure 10. To our surprise, all devices generated traffic equivalent to the traffic generated in state (#6) (one hour) in that 5 mins period. Compared to other previous states we observed the Echo traces to show port access queries – both HTTP and HTTPS – which were not previously observed. Also the Nest Smoke Alarm generated a total of ~35 kB of traffic in that 5 min period after restart, suggesting device restart increases network traffic.



(a) Transport-layer traffic (b) Application-layer traffic

Figure 10: Traffic breakdown by transport- and application-layer protocols for each device when network settings shown in Table 2 are changed from the state (#6) → (#7).

5 RELATED WORK

The challenges posed by the current state of the IoT ecosystem are widespread, providing individual, technological and societal threats. In this context we define a *threat* as the danger resulting from exploitation of vulnerabilities in a system causing potential harm [9, 11]. For example, Bugeja et. al [11] analyse the potential threat agents in home IOT environments and classify them based on their motivations and capabilities. The various threat agents include nation states, terrorists, competitors, and criminals. Their various motivations involve curiosity, personal gain, terrorism, and national interest. To minimise possible threats, deployments will need to provide for various security requirements: authentication, confidentiality, integrity, non-repudiation and availability [2, 39]. Additionally, privacy risks and Human-Data Interaction challenges must be managed, with support for data subjects' rights to control, edit, manage and delete information about themselves, as well as deciding when, how and the extent to which information about them may be communicated to others [34, 47].

Recent years have seen both privacy and security perspectives explored, analysed and presented in many research articles, e.g., [9, 11]. However, there has been relatively little research on how the critical end-to-end services and infrastructure components of the IoT ecosystem could affect scalability, availability, and integrity of these systems. In response, we focus here on understanding the threats linked to the scalability, availability, and integrity that deployment of commodity IoT devices will create.

Abomhara et al [1] analyse IoT threat types and characterise intruders and attacks facing IoT devices and services. Connected devices are found to be rather valuable to cyber-attackers for several reasons: (*i*) most IoT devices operate unattended by humans, so it is easy for an attacker to gain physical access to them; (*ii*) most IoT components communicate over wireless networks without requiring encryption, so attackers might obtain confidential information simply by eavesdropping; (*iii*) most IoT components cannot support complex security schemes due to low power and computing resource capabilities. [2]

Others have recently carried out similar analysis and studies looking into various IoT traffic features, though most focus on privacy and security. Abomhara et al [1] analysed IoT threat types and characterise intruders and attacks facing IoT devices and services. Apthorpe et al [8] discusses privacy leakage from DNS queries and encrypted traffic. Duo et al [14] consider what happens in different scenarios were volumetric data generated between user and cloud services from the compromised devices and app services is exposed. They present a machine learning mechanism to learn the pattern identifying a DDoS attack, install a corresponding filter at the edge of the network, and discuss the amplification factor when services are disrupted due to DDoS attacks.

Dainotti et al [13] discusses IoT network's packet level traffic analysis, inter-packet time and packet size. The analysis done in the paper provides details such as delay, jitter and packet loss. Mahjabin et al [32] discusses impact and scale of DDoS attack. Apthorpe et al [7] suggests four strategies for protecting smart home privacy from home network observers, for example, blocking traffic, concealing DNS, tunnelling traffic, and reshaping traffic. Andradez et al [6] studied the connection time of cars, both spatial and temporal data to find correlation using the time series representation.

Lee et al [30] discuss sequential correlation between DNS queries to show the temporal dependency and vulnerability. Allman et al [4] discuss issues and etiquette concerning the use of shared measurement data. Kumar et al [27] discuss the mis-issuance of security certificate using Zlint, and found only 0.02% of certificate are mis-issued. Lai et al [28] present an algorithm to visualise top DNS server queries. Krohnke et al [26] discuss the impact of location of DNS server on DNS query responses if, for example, it is within only one AS or domain. Hahn et al [22] present interesting work on detection and separation of compressed text from encrypted text using k-nearest neighbour (60%) and 1D convoluted neural network (66.9%).

Finally, Sivanathan et al [42] study IoT traces from a selection of commodity IoT devices. They observe some similar properties to those we report here, but focus analysis on active/inactive periods (finding that most active periods are short), and on clustering observed behaviours among their devices (finding approximately 5 clusters of network behaviour). They use these data and analyses to classify and ultimately identify devices. In contrast, we analyse behaviour in more detail (traffic volumes and periodicity by protocol), and we are particularly interested in the implications for resilience of the built environment with respect to the dependencies we implicitly take by widespread deployment of IoT devices through the network connectivity and Internet services they rely upon.

6 CONCLUSIONS

The traffic capture and analysis presented here adds to the body of work presenting the network-level behaviour of commodity IoT devices. We add a protocol-level breakdown and more detailed analysis of periodicity over a longer time period. This leads to an exploration of the service and infrastructure dependencies that will be taken in "smart" environments when IoT devices are deployed. We thus also present analysis of some of these dependencies from a cloud service and geographical perspective, finding that many devices make use of services distributed across the planet and thus appear dependent on the global network infrastructure even when carrying out purely local actions. Finally, we examine the robustness of device operation when connectivity is disrupted, finding that some devices cease to operate properly without network connectivity (even where their behaviour appears, on the face of it, to require only local information, e.g., the Hive thermostat). Further, they exhibit quite different network behaviours, typically involving significantly more traffic and possibly use of otherwise unobserved protocols, when connectivity is recovered after some disruption. This has implications for device behaviour profiling and firewalling as proposed by, e.g., the IETF's draft Manufacturer Usage Description (MUD) standard [29].

ACKNOWLEDGMENTS

This project was supported by a mini-projects award from the Centre for Digital Built Britain and Innovate UK under Grant 90066, and by EPSRC EP/N028260/1 and EP/R03351X/1.

REFERENCES

[1] Mohamed Abomhara and Geir M. K¯ien. 2015. Cyber Security and the Internet of Things: Vulnerabilities, Threats, Intruders and Attacks. *Journal of Cyber Security* 4 (2015), 4:65–4:88.

[2] Hamoud M. Aldosari. 2015. A Proposed Security Layer for the Internet of Things Communication Reference Model. *Procedia Computer Science* 65 (2015), 95 – 98. International Conference on Communications, management, and Information technology (ICCMIT'2015).

[3] Zigbee Alliance. 2018. An IEEE 802.15.4-based specification for a suite of high-level communication protocols. Retrieved 24-03-2018 from *http://www.zigbee.org/*

[4] Mark Allman and Vern Paxson. 2007. Issues and Etiquette Concerning Use of Shared Measurement Data. In *Proceedings of the 7th ACM SIGCOMM Conference on Internet Measurement (IMC'07)*. ACM, New York, NY, USA, 135–140. *https://doi.org/10.1145/1298306.1298327*

[5] Amazon. 2018. Amazon Echo Dot. Retrieved 24-03-2018 from *https://www.amazon.co.uk/Amazon-Echo-Dot-Generation-Black/dp/B01DFKBL68*

[6] Carlos E. Andrade, Simon D. Byers, Vijay Gopalakrishnan, Emir Halepovic, David J. Poole, Lien K. Tran, and Christopher T. Volinsky. 2017. Connected Cars in Cellular Network: A Measurement Study. In *Proceedings of the Internet Measurement Conference (IMC'17)*. 235–241.

[7] Noah Apthorpe, Dillon Reisman, and Nick Feamster. 2017. Closing the Blinds: Four Strategies for Protecting Smart Home Privacy from Network Observers. (2017). arXiv:1705.06809

[2] Indeed, we observed this ourselves when we struggled to get one of our devices connected because our institution's wireless network policy mandated an enterprise-level security protocol while the device only supported the commonly deployed domestic security protocols.

[8] Noah Apthorpe, Dillon Reisman, Srikanth Sundaresan, Arvind Narayanan, and Nick Feamster. 2017. Spying on the Smart Home: Privacy Attacks and Defenses on Encrypted IoT Traffic. (08 2017).

[9] O. Arias, J. Wurm, K. Hoang, and Y. Jin. 2015. Privacy and Security in Internet of Things and Wearable Devices. *IEEE Transactions on Multi-Scale Computing Systems* 1, 2 (April 2015), 99–109. *https://doi.org/10.1109/TMSCS.2015.2498605*

[10] Arlo. 2018. Arlo smart security camera. Retrieved 24-03-2018 from *https://www.arlo.com/uk/products/arlo/default.aspx*

[11] J. Bugeja, A. Jacobsson, and P. Davidsson. 2017. An analysis of malicious threat agents for the smart connected home. In *Proceedings of the IEEE International Conference on Pervasive Computing and Communications Workshops*. 557–562.

[12] D-Link. 2018. Home Wi-Fi Motion Sensor. Retrieved 24-03-2018 from *https://eu.dlink.com/uk/en/products/dch-s150-motion-sensor*

[13] A. Dainotti, A. Pescape, and G. Ventre. 2006. A packet-level characterization of network traffic. In *Proceedings of the 11th International Workshop on Computer-Aided Modeling, Analysis and Design of Communication Links and Networks*. 38–45. *https://doi.org/10.1109/CAMAD.2006.1649716*

[14] Nhu-Ngoc Dao, Trung V. Phan, Umar Sa ad, Joongheon Kim, Thomas Bauschert, and Sungrae Cho. 2017. Securing Heterogeneous IoT with Intelligent DDoS Attack Behavior Learning. (2017). arXiv:1711.06041

[15] Foobot. 2018. Smart Indoor Air Quality Monitor. Retrieved 24-03-2018 from *https://foobot.io/features/*

[16] Gartner. 2017. Prediction of number of IoT devices. Retrieved 07-02-2017 from *https://www.gartner.com/newsroom/id/3598917*

[17] M. Gebski, A. Penev, and R. K. Wong. 2006. Protocol Identification of Encrypted Network Traffic. In *Proceedings of the IEEE/WIC/ACM International Conference on Web Intelligence(WI’06)*. 957–960. *https://doi.org/10.1109/WI.2006.139*

[18] Nam K. Giang, Rodger Lea, Michael Blackstock, and Victor C. M. Leung. 2016. On Building Smart City IoT Applications: A Coordination-based Perspective. In *Proceedings of the 2nd International Workshop on Smart Cities (SmartCities’16)*. ACM, New York, NY, USA, Article 7, 6 pages. *https://doi.org/10.1145/3009912.3009919*

[19] Google. 2018. Google Home Mini. Retrieved 24-03-2018 from *https://store.google.com/product/google_home_mini*

[20] GraphViz. 2018. Open-source Graph Visualization Software. Retrieved 24-03-2018 from *https://www.graphviz.org/*

[21] Bluetooth Special Interest Group. 2018. Bluetooth: a wireless technology standard for exchanging data over short distances. Retrieved 24-03-2018 from *https://www.bluetooth.com/*

[22] Daniel Hahn, Noah Apthorpe, and Nick Feamster. 2018. Detecting Compressed Cleartext Traffic from Consumer Internet of Things Devices. (2018). arXiv:1805.02722

[23] Hive. 2018. Hive Hub. Retrieved 24-03-2018 from *https://www.hivehome.com/products/hive-hub*

[24] Hive. 2018. Hive Motion Sensor. Retrieved 24-03-2018 from *https://www.hivehome.com/products/hive-motion-sensor*

[25] BI Intelligence. 2018. Prediction of number of IoT devices. Retrieved 27-02-2018 from *http://uk.businessinsider.com/the-internet-of-things-2017-report-2018-2-26-1*

[26] Lars Kröhnke, Jelte Jansen, and Harald Vranken. 2018. Resilience of the Domain Name System: A case study of the .nl-domain. *Computer Networks* 139 (2018), 136 – 150. *https://doi.org/10.1016/j.comnet.2018.04.015*

[27] D. Kumar, Z. Wang, M. Hyder, J. Dickinson, G. Beck, D. Adrian, J. Mason, Z. Durumeric, J. A. Halderman, and M. Bailey. 2018. Tracking Certificate Misissuance in the Wild. In *Proceedings of the IEEE Symposium on Security and Privacy (SP)*. 288–301. *https://doi.org/10.1109/SP.2018.00015*

[28] Qingnan Lai, Changling Zhou, Hao Ma, Zhen Wu, and Shiyang Chen. 2015. Visualizing and characterizing DNS lookup behaviors via log-mining. *Neurocomputing* 169 (2015), 100 – 109. *https://doi.org/10.1016/j.neucom.2014.09.099*

[29] Eliot Lear, Ralph Droms, and Dan Romascanu. 2018. *Manufacturer Usage Description Specification*. Internet-Draft draft-ietf-opsawg-mud-25. Internet Engineering Task Force. *https://datatracker.ietf.org/doc/html/draft-ietf-opsawg-mud-25* Work in Progress.

[30] Jehyun Lee and Heejo Lee. 2014. GMAD: Graph-based Malware Activity Detection by DNS traffic analysis. *Computer Communications* 49 (2014), 33 – 47. *https://doi.org/10.1016/j.comcom.2014.04.013*

[31] LightwaveRF. 2018. Lightwave RF. Retrieved 24-03-2018 from *https://lightwaverf.com/*

[32] Tasnuva Mahjabin, Yang Xiao, Guang Sun, and Wangdong Jiang. 2017. A survey of distributed denial-of-service attack, prevention, and mitigation techniques. *International Journal of Distributed Sensor Networks* 13, 12 (2017), 1550147717741463. *https://doi.org/10.1177/1550147717741463*

[33] IHS Markit. 2017. Prediction of number of IoT devices. Retrieved 24-10-2017 from *https://technology.ihs.com/596542/number-of-connected-iot-devices-will-surge-to-125-billion-by-2030-ihs-markit-says*

[34] Richard Mortier, Hamed Haddadi, Tristan Henderson, Derek McAuley, Jon Crowcroft, and Andy Crabtree. 2016. *Human-data interaction* (2nd ed.). Vol. The Encyclopedia of Human-Computer Interaction. Interaction Design Foundation, Chapter 41. arXiv:https://bit.ly/encyclopedia-hdi *https://www.interaction-design.org/literature/book/the-encyclopedia-of-human-computer-interaction-2nd-ed/human-data-interaction* Mads Soegaard and Rikke Friis Dam (eds.).

[35] Nest. 2018. Nest Protect smoke and CO alarm. Retrieved 24-03-2018 from *https://nest.com/uk/smoke-co-alarm/overview/*

[36] Netgear. 2018. Wireless Dual Band Gigabit Router. Retrieved 24-03-2018 from *https://www.netgear.com/support/product/WNDR3700v2.aspx*

[37] NTP. 2018. NTP Pool Project. Retrieved 24-03-2018 from *http://www.pool.ntp.org/en/*

[38] Parrot. 2018. Parrot POT. Retrieved 24-03-2018 from *https://www.parrot.com/uk/connected-garden/parrot-pot#parrot-pot*

[39] J. Ren, H. Guo, C. Xu, and Y. Zhang. 2017. Serving at the Edge: A Scalable IoT Architecture Based on Transparent Computing. *IEEE Network* 31, 5 (2017), 96–105. *https://doi.org/10.1109/MNET.2017.1700030*

[40] RFID. 2018. Radio Frequency Identification. Retrieved 24-03-2018 from *https://en.wikipedia.org/wiki/Radio-frequency_identification*

[41] Semtech. 2018. Lora: a patented wireless data communication technology. Retrieved 24-03-2018 from *https://lora-alliance.org/*

[42] Arunan Sivanathan, Daniel Sherratt, Hassan Habibi Gharakheili, Adam Radford, Chamith Wijenayake, Arun Vishwanath, and Vijay Sivaraman. 2017. Characterizing and Classifying IoT Traffic in Smart Cities and Campuses. In *Proc. IEEE INFOCOM Workshop on Smart Cities and Urban Computing*.

[43] Bilal Succar. 2009. Building information modelling framework: A research and delivery foundation for industry stakeholders. *Automation in Construction* 18, 3 (2009), 357 – 375. *https://doi.org/10.1016/j.autcon.2008.10.003*

[44] TCPDump. 2018. A command-line packet analyzer. Retrieved 24-03-2018 from *https://www.tcpdump.org/*

[45] Smart Times Technology. 2018. Mini Smart Bluetooth Tracker. Retrieved 24-03-2018 from *https://www.1dayfly.com/cms_img/track_en_trace_manual.pdf*

[46] TP-Link. 2018. Wi-Fi Smart Plug with Energy Monitoring. Retrieved 24-03-2018 from *https://www.tp-link.com/uk/products/details/cat-5258_HS110.html*

[47] Alan F. Westin. 1967. *Privacy and Freedom.* New York: Atheneum.

[48] Wireshark. 2018. Open-source Network protocol analyzer. Retrieved 24-03-2018 from *https://www.wireshark.org/*

[49] Xiaomi. 2018. Mi Band 2. Retrieved 24-03-2018 from *http://www.mi.com/en/miband2/*

[50] Zensys. 2018. Zwave: a wireless communications protocol used primarily for home automation. Retrieved 24-03-2018 from *http://www.z-wave.com/*

[51] H. Zhang, J. Li, B. Wen, Y. Xun, and J. Liu. 2018. Connecting Intelligent Things in Smart Hospitals using NB-IoT. *IEEE Internet of Things Journal* (2018), 1–1. *https://doi.org/10.1109/JIOT.2018.2792423*